Nov 84

The program is WIZARD (C) 1984 by Progressive Peripherals & Software

TYPE OF PROTECTION: The disk is littered with errors. When we attempt to examine the disk with a track and sector editor, we find that accessing various tracks will cause the drive to read endlessly (ex. TRACKS 2 and 3). We also discover that track 18/0 has been modified to make the directory unlistable.

HOW TO COPY THE DISK: This program uses a non-standard format that most nibble copy programs cannot handle. You may be sure that a copier will soon be developed to handle a program of this type.

HOW TO MAKE A WORKING COPY WITHOUT ERRORS.

PROBLEMS:
1). We must modify 18/0.
2). Code is stored under the kernal.
3). We must find a proper entry point.
4). We must write a loader program to bring our programs into memory and execute the proper SYS command.

SOLUTIONS:
Let's get to work. This program is a bit complex, but certainly not impossible. Through the unprotection of this program, we will add some useful techniques to our bag of tricks.

1). First, make a copy the entire disk without errors.

2). With your track and sector editor executed, alter 18/0 by moving the name over two bytes. Go ahead and use WIZARD for the name, but then go to HEX mode and fill in the rest of the spaces from the end of the name to the ID (WZ) with A0. This will allow us to get a proper listing of the directory.

3). Now that you have a proper directory, file copy all but the programs listed below to a newly formatted disk. This is the disk we will be working with.

WIZ
LODR
CHRW
SPRW
M.L.

4). Load and Run the program from the original disk. When the option screen appears, insert your cartridge based machine language monitor. Reset your computer with your cartridge switch OFF. We leave the switch off so that we may preserve the coded stored by the program at 8000. You should perform this reset before the program goes into the DEMO or PLAY modes. You should get the familiar blue screen showing 30719 BYTES FREE. Remove the cartridge.

5). Load and execute LLMON with SYS8192.

6). Using the I command, scroll through the memory. The first code we find is at $1000. At 11F0, we will find the code to load the programs that we deleted from our copy disk. This is the code from the LODR program. If we use the D command, we will find numerous kernal routines. Among them, is a routine to re-format a copy disk, when it is detected by the program. Have no fear, we'll skip that part. When we return to interpret mode, we see the preparatory screens used before we enter the option screen. After a little experimentation, we find that the code at $198C will load in a program called SCOR. This is the program that will keep track of the high scores earned in the game. When we run this section of code with a G 198C (with our copy disk in the drive), we find that we have a working version of the program. There's only one hitch, we have some valuable code stored under the KERNAL. This code controls the sprites. We'll solve that problem in a moment.

7). S "WIZARD",08,1000,CFFF - Do not exit the monitor yet. We will now write a short program to access the code stored under the kernal.

8). Through LLMON, load in HIMON. Once loaded, execute HIMON with G C000. Using the F command, fill the memory from 1000 to 9FFF with 00. This will clean things up for our next operation. We will construct our routine at 1000. Using the M command, type in the following:

```
.:1000 78 A9 35 85 01 A9 00 85
.:1008 FC 85 FE A9 E0 85 FD A9
.:1010 20 85 FF A0 00 B1 FC 91
.:1018 FE C8 C0 00 D0 F7 E6 FF
.:1020 E6 FD A5 FD D0 ED A9 37
.:1028 85 01 58 00 00 00 00 00
```

Make sure you have typed in the code correctly. Don't forget to type return after each line. When we disassemble our program at $1000, we find the following.

```
1000  SEI               1017  STA ($FE),Y
1001  LDA #$35          1019  INY
1003  STA $01           101A  CPY #$00
1005  LDA #$00          101C  BNE $1015
1007  STA $FC           101E  INC $FF
1009  STA $FE           1020  INC $FD
100B  LDA #$E0          1022  LDA $FD
100D  STA $FD           1024  BNE $1013
100F  LDA #$20          1026  LDA #$37
1011  STA $FF           1028  STA $01
1013  LDY #$00          102A  CLI
1015  LDA ($FC),Y
```

This program will set the interrupt, which will keep us from locking up, and transfer the code stored under the KERNAL to $2000. Save this program to a disk and keep it for future use. Now execute the program with G 1000. Using the I command, scroll through the code at 2000. Save this code with S "KERNAL",08,2000,4000. We may now exit the monitor.

9). We have a slight problem with our kernal program. When loaded, KERNAL will locate at $2000. The program is expecting to find it at $E000. We could write a loader program to relocate this code, or we could do it the easy way. With a track and sector editor, we can locate where the first block of the KERNAL program is stored on the disk. Go to TRACK 18 and locate KERNAL in the directory. It will be the last program in the directory and should be located on TRACK 6 SECTOR 0. Bytes 3 and 4 will tell you where the program is located. Once you have the location, go to that TRACK 6/0. Byte 3 will contain our 20. If we change this to E0, the program will relocate to $E000 when it is loaded. The block should read as follows:

```
060A0020
```

060A - Location of the next block of the program.
0020 - Low byte - High byte of the memory address for this program. Change the 20 to an E0.

10). All that's left is to write a loader that will load in the two programs and SYS to 6540 (198C HEX). The following program will do the job for us:

```
10 PRINT"[CLR][3 DOWN]LOAD"CHR$(34)"WIZARD"CHR$(34)",8,1"
20 PRINT"[4 DOWN]NEW"
30 PRINT"[2 DOWN]LOAD"CHR$(34)"KERNAL"CHR$(34)",8,1"
40 PRINT"[4 DOWN]NEW"
50 PRINT"[2 DOWN]SYS6540[HOME]"
60 POKE198,5:FORI=0TO4:POKE631+I,13:NEXT
```

SAVE this program to your copy disk (SAVE "WIZARD LOADER",8) and then RUN it.

11). YOU'RE DONE.

FOR THE IMPATIENT AMOUNG US.

When run, our unprotected program will take 30 seconds longer to load than the original. We may shorten the load time by saving the code out in pieces. The sections necessary for proper execution are as follows:

1000 - 2000
5800 - 9FFF
C000 - CFFF

With these sections and the kernal section we saved earlier, the program will run properly. Don't forget to change the loader program to accommodate the four programs.

The program is DECATHLON, COPYRIGHT 1983-84 ACTIVISION.

TYPE OF PROTECTION: This program checks track 18 sector 9 for an error 23 CHECKSUM ERROR IN DATA. If this error is present, the program will run properly.

HOW TO COPY: Copy the disk with any good copy program and place an error 23 on the disk at 18/09. You may also use a nibble copy program and let it put the errors on for you.

HOW TO MAKE A WORKING COPY WITHOUT ERRORS ON THE DISK.

We will combine many of the techniques illustrated in the F.P.M. to unprotect this program. The problems we will encounter are as follows:

PROBLEMS:
1). Getting past the error checking.

2). The program resides in cartridge memory, producing an auto-start and preventing a normal reset.

3). A portion of the program resides under basic.

SOLUTIONS:
1). Let's get past the error protection first. LOAD and RUN the program. With your cartridge switch (P.61) off, insert your cartridge based machine language monitor. Reset your computer with your switch off. This will return us to basic. Remove your cartridge monitor.

2). LOAD and execute LLMON with SYS 8192.

3). Using the M command, go to location 0001 and change the 37 to a 36. This will flip out the basic interpreter, allowing us to access the code underneath.

```
.:0000 2F 36 20 3E 32 00 23 00
```

4). S "DECATHLON",08,8000,C400

5). LOAD"DECATHLON",8,1

6). Press RESTORE to activate the program.

7). YOU'RE DONE.

This is a good time to explain the function of the non-maskable interrupt (NMI). There are three types of interrupts: IRQ (interrupt request), NMI (non-maskable interrupt), and the BRK (break instruction).

The interrupt vector goes about it's work sixty times a second. Among it's chores is updating the clock, checking the RUN/STOP key, handling keyboard input, and other functions necessary for the proper functioning of the computer. With proper preparation, we may change the interrupt vector so that it will suspend normal operations and work on the task we request. Such a request would begin with SEI (set interrupt disable).

The BRK (break) instruction will also suspend the interrupt vector. This instruction is normally used to debug machine language routines.

The NMI is the function that concerns us in this program. On the VIC-20 and Commodore 64, the NMI is used for the RESTORE key and for RS-232 communications. When the RESTORE key is pressed, the computer will suspend operations and proceed to the warm-start vectors stored by the programmer. This is why we are able to interrupt games stored in the cartridge area. Upon power-up or a system reset, the computer will check the cartridge area (8000). If the CBM80 is present, the computer will divert the program to the vectors (entry point) stored there. Remember, the first two vectors are the cold start vectors and the next two are the warm-start. Remember also that these addresses are stored in reverse order. The warm-start vectors are used when the RESTORE key is pressed.

In DECATHLON, a warm start is provided through the program. With LLMON activated, let's take a look at $8000.

```
 Cold-start  Warm-start  CBM80
 55 80       69 80       C3 C2 CD 38
```

If we load in our unprotected version of the program and enter SYS 32873 (HEX 8069), we will enter the warm-start vector. Fortunately we do not have to remember the address, because the RESTORE key will divert us to this address through the NMI.

This is why we had to use our cartridge based machine language monitor to interrupt this program. When we insert the cartridge monitor and perform a reset, the program sends control to the cartridge. With the switch off, the computer recognizes that a cartridge is present, but will not perform the auto run until the switch is turned on. When the cartridge is removed, the original program code is intact.

Another way to defeat the CBM80 is to search the disk for the CBM80 and change one of the bytes (i.e. change the '8' to '0'). When the reset button is pressed the computer will reset normally. All five bytes of the CBM80 must be present in order for control to be turned over to the vectors. Remember to change the CBM80 back to normal before saving the program to disk.

When run, many programs will alter their code to check the error channel. They will then store these values within the program. If the correct value is not found, the program will not execute properly. These programs are a bit more difficult to unprotect, but with a little effort the result is the same. Programs of this type become easier as we gain experience working with them.

The program is NIGHT MISSION PINBALL, Copyright, Sublogic Corporation., 1982

TYPE OF PROTECTION: The program is stored on the disk in user files. The program checks track 03 for an error 21.

HOW TO COPY: Use a good copy program and place an error 21 on Track 03. You may also use a nibble copy program and let it put the errors on for you.

HOW TO MAKE A WORKING COPY WITHOUT ERRORS ON THE DISK.

PROBLEMS:

1). The program is stored in user files making it difficult to examine the code.

2). The program will check the error channel after a program option is selected.

3). A portion of the program is stored under basic.

4). An entry point must be found.

5). We must write a small section of code to flip out basic and then jump to the proper entry point.

SOLUTIONS:

1). Load and execute HIMON.

2). Using the M command at 0001 change the 37 to a 36. This will flip out the basic interpretor. Fill the memory from 0800-BFFF with 00. Change location 01 back to a 37.

3). Using the X command, exit to basic and load and run the program from the original disk.

4). When the option menu appears, choose regular mode, but do not play the game.

5). Reset your computer.

6). Activate HIMON with SYS 49152.

7). Using the M command at 0001, change the 37 to a 36. This will allow us to save out the code stored under basic.

8). Disassemble the code at 093A. This is where the program will check the error channel. Since we have already run this section of code and have the values we need stored in memory, we will eliminate this section of code. This will keep the program from overwriting the values we need for proper execution. Using the M command at 093A, change the A9 to a 60 (RTS).

9). Now we have to write the boot to flip out basic and jump to the entry point. We store this code in a section of memory that is not being used by the program. Using the M command, we find that 0A00 is available. Change the first 7 bytes at 0A00 to the following: A9 36 85 01 4C 61 08. When we disassemble the code, we find the following:

```
LDA #$36   - Load the accumulator with the value of 36.
STA $01    - Store this value at 01 - This will flip out basic.
JMP $0861 - This will take us back into the program.
```

10). Using the I command, locate the ending address of the program. You will find that point at B098.

11). S "NIGHTSYS2560",08,0800,B098

12). YOU'RE DONE.

When you load this program in, you must SYS to 2560. This is the decimal equivalent for 0A00, which is where we stored our program to flip out basic. It's always a good idea to save a SYS location in the program name in case you forget the location.

The only real trick to this program is finding the entry point. This is really a matter of experimentation. We searched for a spot that would put us into the menu screen.

The program is THE QUEST, Copyright Penquin Software, Inc., 1984

TYPE OF PROTECTION: This program requires and error 23 on track 2 and an error 27 on track 3. If the errors are present, the program will execute properly.

HOW TO COPY: Copy the disk with any good copy program and place an error 23 on track 2 and an error 27 on track 3. You may also use a nibble copy program and let it put the errors on for you.

HOW TO MAKE A WORKING COPY WITHOUT ERRORS ON THE DISK.

PROBLEMS:

1). The program that contains the protection (AUTO) is an auto boot, making the code difficult to examine. Page 52 of the P.P.M will give you some hints on auto load programs.

2). The program loads the accumulator (LDA) with the values it will later compare (CMP).

SOLUTIONS:

1). Make a copy of the original disk without errors.

2). So that we may examine the code without the program taking over control of the computer, we will load the program in the area of memory normally used for BASIC programs. Load "AUTO",8. Do not use ,8,1. Now execute HIMON and disassemble the code at 089E. Keep in mind that we are examining a program that would nomally reside at 0300. At 089E, we find LDA #$33 and STA at $03EF. If we we disasssemble 08EF (03EF), we find a CMP #$37. At 08AF, we find LDA #$37 followed by a STA at $03EF. Looking further, we find a CMP #$32 at 08E8 and CMP #$37 at 08EF. The program will check the channel four times. We must change the LDA's and CMP's to 30 to unprotect this program.

3). The easiest way to alter this program is to make the corrections on the disk. Let's get to the program through a track and sector editor. At 18/01, we find that the program we are interested in (AUTO) is stored at 17/03. If we go to 17/03, we find the first block of the file. In ASCII mode, we find the B-R for tracks two and three. If we go to HEX mode, we will find our A9's and C9's. Let's remember that what we see stored on the disk is similar to the M command in a monitor. LDA is stored as A9 and CMP is stored as C9.

4). Look for A933. This will load the accumulator (LDA) with 33. Change the 33 to a 30. It is not necessary to change the A9. Now look for the A937 and change the 37 to a 30. That takes care of the loads, now let's change the compares.

5). Find the C932 and change the 32 to a 30. With that accomplished, find the C937 and change the 37 to a 30.

6). With the four changes made, use your write function to write the changes to the disk.

7). YOU'RE DONE.

There are track and sector editors on the market that include an assembler function. You may find a program of this type helpful with auto load programs. If you find it difficult to interpret the code found on the disk, a program of this type is what you need. Refer to the New Products Section for a review of such a program. You may also use the U1 and U2 programs included on the P.P.M. disk. These programs will require a few extra steps, but they will do the job. Page 79 of the P.P.M. contains the information necessary to use these programs.

SPECIAL FEATURE FOR OWNERS OF PROMENADE.
For the advanced programmer who would like to enhance the capabilities of Hessmon (TM), we will describe the process of eliminating some of the resets below $0400. This will allow you more flexibility in the examination of programs that store to the vectors normally reset by Hessmon. Change the code to the following:

```
8E07 A9 E7 85 01 EA EA EA EA
8E25 EA EA EA A9 8E EA EA EA
8E2A EA EA EA A9 01 8D 20 D0
```

Hessmon (TM), Copyright 1982, Human Engineered Software.

NEW PRODUCTS REVIEWED BY C.S.M.

DISKMAKER (TM) FROM BASICS $49.95

Diskmaker (TM) is a machine language copy utilty. We have had many requests for a program that could handle a specific track or range. This is the program for you. It is fully automatic and will copy up to 40 tracks. Diskmaker (TM) is capable of copying many of the newest copy p-otected programs, including Centinel and Zaxxon. This copy program is the closest to a true nibble copier we have seen to date.

INSIDE COMMODORE DOS (C) Datamost 1984 $19.95

Richard Immers and Gerald G. Neufeld have done an outstanding job of explaining the inner working of the disk drive. This book includes chapters on data storage, recovering damaged disks, understanding drive operation, and backing up protected diskettes. With Inside Commodore Dos, The Anatomy of The Disk Drive, and Omni Clone's Source Code, you should have no problem mastering the disk drive.

PEEK A BYTE (C) Slaymaker 1984 $29.95

Peek A Byte is an outstanding track and sector editor. Dr. Slaymaker has developed a program that provides all the features we need in one program. When you buy this program, you've got it all. The following will give you an idea of the capabilities of this program:

1. Resident program occupying $C000-$CFFF
2. Examine and edit almost any memory location in ASCII, screen code, or HEX.
3. May be used with basic programs or a machine language monitor.
4. Move disk drive memory into computer memory.
5. Check for disk errors.
6. Help screens
7. Complete User's manuel.
8. Auto disk read
9. File follower
10. Compatible with MSD
11. Disassemble disk data in any buffer or page and dump to printer.

If you're tired of trying to remember which program does what, this is the program for you. Peek A Byte is available through Quantum Software, P.O.Box 12716, Lake Park, Fl. 33403.

ATTENTION AUTHORS:

If you have a program that is related to copy protection, C.S.M. will review it in the NEWSLETTER. We will also be willing to sell programs we find particularly useful.

NOV. 1984

ALL ITEMS IN STOCK FOR IMMEDIATE SHIPMENT

1541 DISK DRIVE ALIGNMENT PROGRAM $44.95
See Oct. 84 Compute! Gazette for review

PROGRAM PROTECTION MANUAL FOR THE C-64 29.95

PROGRAM PROTECTION NEWSLETTER 35.00

A monthly newsletter covering program protection. 3 to 5 different program protection schemes will be covered each month. A new section on the DOS and other special interests will also be included.

CARDO 5 SLOT EXPANSION BOARD FOR THE C-64 60.00
INCLUDES A RESET BUTTON AND THE CARTRIDGE SWITCH

ADVANCED ML BOOK FOR THE C-64	(ABACUS)	19.95
ANATOMY OF THE 64	(ABACUS)	19.95
ANATOMY OF THE 1541	"	19.95
MACHINE LANGUAGE BOOK OF THE 64	"	14.95
TRICKS AND TIPS FOR THE 64	"	19.95

1541 MAINTENANCE MANUAL (GOSUB) 29.95

MSD SUPER DISK DRIVE DUAL 580.00

Each unit fully tested and ready to run. Before we send out the MSD, we verify that is in tip-top shape. Don't take a chance on buying a unit that is not FULLY checked out. (DURING THE PAST MONTH WE HAVE REJECTED SIX (6) DRIVES AND SENT THEM BACK TO MSD FOR REPLACEMENT).

PROMEMADE 99.50

THIS IS THE EPROM PROGRAMMER THAT WE USE HERE!! USE IT TO MAKE YOUR OWN OPERATING SYSTEM CHIPS, CARTRIDGE PROGRAMS OR A NEW DOS FOR THE DISK DRIVE. SOFTWARE INCLUDED ON CASSETTE.

DATA ERASE $34.95

This will erase all U.V. erasable EPROMS. A compact and easy to use unit that will erase 2 EPROMs in 15 minutes or less.

BOARDS, SOCKETS AND EPROMS:

PCC2	2 CHIP BOARD	$6.95
PCC4	4 CHIP BANK SWITCHED BOARD	19.95
PCVH2	PLASTIC HOUSING FOR ABOVE BOARDS	2.95
LP28	28 PIN SOCKET FOR EPROM	.59
ZIF28	ZERO INSERTION FORCE 28 PIN SOCKET	4.75
2732	EPROM (USE IN CARTRIDGE 4K)	5.00
2764	EPROM (USE IN CARTRIDGE 8K OR IN DISK DRIVE W/AD)	7.50
AD	USE THIS ADAPTER TO USE THE 2764 EPROM IN THE DISK DRIVE OR IN THE COMPUTER. IF YOU DON'T USE THE ADAPTER USE 68764 EPROM	8.95
HESMON	(R) ML MONITOR ON A CARTRIDGE. This version also includes a DOS WEDGE and PROMOS software on the cartridge.	25.00

DISKMAKER BY BASIX $49.50

INSIDE COMMODORE DOS BY DATAMOST 19.95

CARTRIDGE CRACKER II (NEW SOFTWARE--NEW PRICE) 49.50

THIS WILL 'BREAK' 99% OF ALL CARTRIDGES ON THE MARKET TODAY! IT WILL DOWN LOAD THE NEW VERSION OF THE PROGRAM TO DISK AND WRITE AN AUTO BOOT LOADER FOR IT. IT WILL ALSO TELL YOU WHAT CODE IT CHANGED IN ORDER TO 'BREAK' THEM. THE UNIT WILL DO 'COMMODORE' (R) CARTRIDGES (INCLUDING SIMONS), PARKER BROS (R), EPYX (R) AND MORE. IT WILL ALSO FIX THE ELECTRONIC ARTS (R) DISK PROGRAMS AND DO MORE OF THEM THAN ANY OTHER PROGRAM ON THE MARKET. THE PROGRAM ALSO CONTAINS SUPER SAVER - DISK TO DISK AND DISK TO TAPE FILE COPYING. The new software for the CARTIRDGE CRACKER may be ordered directly from BEAR TECHNOLOGY ($5.00 + $3.00 Shipping).

XETEC PRINTER INTERFACE 59.95

THIS INTERFACE WILL TURN YOUR GEMINI, DELTA OR EPSON PRINTER INTO A CORRESPONDANCE QUALITY PRINTER!!! THIS FLYER WAS PRINTED ON A GEMINI 10X WITH A XETEC PRINTER INTERFACE IN THE CORRESPONDENCE MODE.

THREE MINUTE COPY PROGRAM.....OMNICLONE 35.00

NOT-PROTECTED AND FULLY DOCUMENTED.
AT LAST, A COPY PROGRAM THAT WILL NOT ONLY COPY MOST DISKS (ERRORS AND ALL), BUT DOCUMENTATION (SOURCE CODE) TO GO WITH. THIS WILL COPY THE DISK AND AUTOMATICALLY PLACE MOST ERRORS (20,21,22,23,27 & 29) ON THE DESTINATION DISK. THREE PASSES, THREE MINUTES. DON'T CONFUSE THIS PROGRAM WITH SPEED COPY (GERMAN COPY PROGRAM) THAT ONLY DOES ERROR 23; THIS DOES THEM ALL. FULL DOCUMENTATION IS INCLUDED SO THAT YOU CAN MAKE THIS THE BEST COPY PROGRAM OF ALL. LEARN ABOUT HALF TRACKING, HIGH SPEED DATA TRANSFER AND MUCH, MUCH MORE... PLUS, WHEN YOU MODIFY THIS PROGRAM SEND YOUR UPDATES TO US, WE WILL INCLUDE THEM IN FUTURE ISSUES OF THE NEWSLETTER.

ALL ITEMS PLUS SHIPPING. $2.50 US on most items. $8.00 for MSD dual

* *

SPECIAL PURCHASE

DISK DRIVE BELT-- post paid in the US & Canada $10.00
This is the belt that spins the disk. It goes between the drive motor and the flywheel. Many times the belt wears out or is installed improperly from the factory. Use this HEAVY DUTY replacement to bring your drive back up to specs. This belt is made by a commercial and industrial disk drive manufacturer and should outlast the belt supplied as original equipment.

C.S.M. SOFTWARE
P. O. BOX 563
CROWN POINT, IN. 46307
219-663-4335

ERROR 29 ID MISMATCH MAKER

```
100 OPEN15,8,15,"IO:"

110 INPUT"TRACK #";T

120 INPUT "ASCII FOR CHAR 1";I

130 INPUT "ASCII FOR CHAR 2";D

140 ID$=CHR$(I)+CHR$(D)

150 OPEN5,8,5,"#":PRINT#15,"U1";5;0;T;0:CLOSE5

160 PRINT#15,"M-W";CHR$(0)CHR$(6)CHR$(3)CHR$(76)CHR$(199)CHR$(250)

170 PRINT#15,"M-W";CHR$(81)CHR$(0)CHR$(1)CHR$(T)

180 PRINT#15,"M-W";CHR$(18)CHR$(0)CHR$(2)ID$

190 PRINT#15,"M--W";CHR$(3)CHR$(0)CHR$(1)CHR$(208)

200 FORT=0TO3000:NEXT

210 PRINT#15,"U;":PRINT#15,"IO:"

220 CLOSE 15

READY.
```

This routine will format one track of the disk with the new ID. You must use a different ID on the track to be formatted. When the program runs the ID of each track and sector is read into memory and compared with the ID of track 18, sector 0. If they match everything is OK. If not, there is an ID mismatch, error 29 generated.